

AD-A040 147

COLORADO STATE UNIV FORT COLLINS DEPT OF MATHEMATICS  
ADAPTIVE PIECEWISE POLYNOMIAL UNIFORM APPROXIMATION. (U)  
MAR 77 J A HULL

F/G 12/1

AF-AFOSR-28-78-76

UNCLASSIFIED

AFOSR-TR-77-0643

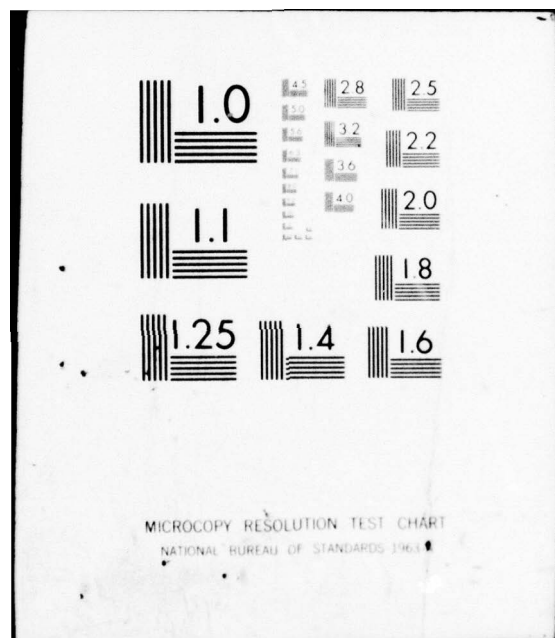
NL

1 OF 1  
AD  
A040147



END

DATE  
FILMED  
6-77



3  
NW

ADA 040 147

ADAPTIVE PIECEWISE POLYNOMIAL  
UNIFORM APPROXIMATION

by

Joseph A. Hull\*

Department of Mathematics  
Colorado State University  
Fort Collins, Colorado 80523

Approved for public release;  
distribution unlimited.

DDC  
RECEIVED  
JUN 3 1977  
A -

AD No. —  
DDC FILE COPY

\*Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-76-2878.

AFSC - 19-12-7b

AFSC 19-12-7b

AFSC 19-12-7b


AFSC 19-12-7b

AFSC 19-12-7b

AFSC 19-12-7b

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)  
NOTICE OF TRANSMITTAL TO DDC  
This technical report has been reviewed and is  
approved for public release IAW AFR 190-12 (7b).  
Distribution is unlimited.  
A. D. BLOSE  
Technical Information Officer

DDC LIFE COPY  
AFSC 19-12-7b

ACCESSION FOR	
NTIS	White Section <input checked="" type="checkbox"/>
DTIC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
AUTHENTICATION	
DISTRIBUTION AVAILABILITY CODES	
Dist.	MAIL, 306, & SPECIAL
	

## CHAPTER I

### Introduction

In this paper we shall develop two algorithms for curve fitting which will be adaptive in nature using results from uniform approximation theory. The organization of this paper is as follows: in this section we shall review some basic results from uniform approximation theory, uniform approximation theory with side conditions, and the Remes algorithm; in the second section we shall give our curve fitting algorithms; and in the last section we shall report on our numerical testing of our algorithms.

Let  $X$  be a compact topological space. Let  $C(X)$  denote the family of all real-valued continuous functions defined on  $X$ . As is well known,  $C(X)$  is a complete normed linear space under the supremum (Chebyshev, uniform, max, infinity) norm

$$\|f\| = \max_{x \in X} |f(x)|, f \in C(X).$$

Let  $S$  be a subset of  $C(X)$  and  $f \in C(X)$  be fixed. Consider

$$\inf_{p \in S} \|f - p\| = d.$$

If there is a  $p^* \in S$  such that  $\|p^* - f\| = d$ , then  $p^*$  is called a best approximation to  $f$  from  $S$ . For a given subset  $S$ , the following questions may be posed.

1. Does a best approximation exist for a particular  $f \in C(X)$ ?

For each  $f \in C(X)$ ?

2. If a best approximation does exist, is it unique?
3. If a best approximation exists, is there an efficient way of computing it?
4. Does a best approximation depend continuously on the function being approximated? (That is, will small errors inevitably introduced when calculating best approximations prevent an algorithm from succeeding in converging to something close to a best approximation?)

In order to answer these questions, we restrict our attention to the following  $X$  and  $S$ . Let  $X$  be a compact subset of an interval on the real line, and  $S$  be a (finite dimensional) Haar subspace of  $C(X)$ . That is, if  $S$  has dimension  $n$ , then the only function in  $S$  having  $n$  or more zeros is the zero function.

For this particular  $X$  and  $S$ , all of the above questions have been answered in the affirmative. For theorems and proofs regarding questions 1, 2 and 4, see for example [3]. We now address ourselves specifically to question 3. The following theorem characterizing best uniform approximations will prove useful.

CHARACTERIZATION THEOREM [3, p. 75]. If  $S$  has dimension  $n$ , then  $p^* \in S$  is a best approximation to  $f \in C(X) \sim S$  if and only if there are  $n + 1$  points,  $x_1, \dots, x_{n+1}$  in  $X \subset [a, b]$  with  $x_i < x_{i+1}$ , ( $i=1, \dots, n$ ) such that  $f(x_i) - p(x_i) = (-1)^i \lambda$ , and  $|\lambda| = \|f - p^*\|$ .

Given any  $f \in C(X) \sim S$ , then any set of  $n + 1$  distinct points from  $X$  determines uniquely a  $p \in S$  and a  $\lambda \in \mathbb{R}$  so that  $p(x_i) + (-1)^i \lambda = f(x_i)$  since  $S$  is a Haar subspace (see [3]). The characterization

theorem and the above observation allow us to view the problem of finding the best approximation to  $f \in C(X)$  as finding a set (called an extreme or alternating set) of  $n + 1$  points,  $\{x_1, \dots, x_{n+1}\} \subset X$  where  $x_i < x_{i+1}$ ,  $i = 1, \dots, n$  so that the  $p$  and the  $\lambda$  determined by the extreme set have the property that  $|\lambda| = \|f - p\|$ . In this case,  $p$  must be the best approximation to  $f$  from  $S$ .

We now describe Remes second (or multiple exchange) algorithm which calculates the best approximation to  $f \in C(X) \sim S$  by finding an extreme set for  $f$ . We assume that  $f$  is fixed and that an arbitrary reference set  $R = \{x_1, \dots, x_{n+1}\}$  of  $n + 1$  distinct points from  $X$  has been chosen so that  $x_i < x_{i+1}$ ,  $i = 1, \dots, n$ , with the added assumption that the  $\lambda$  determined by this reference set is not zero. Since  $f \notin S$ , and  $S$  has only dimension  $n$ , such a set may always be readily found.

We begin each iteration by computing the  $p$  and  $\lambda$  associated with the reference set  $R$ . Let  $\sigma(x) = f(x) - p(x)$ . Since  $\sigma(x_i) = (-1)^i \lambda$  with  $\lambda \neq 0$ ,  $\sigma(x)$  changes sign from one extreme point to the next. If  $[x_i, x_{i+1}] \subset X$ , then since  $f$  and  $p$  are both continuous  $\sigma$  has a zero in  $[x_i, x_{i+1}]$ , call it  $z_i$ . If  $[x_i, x_{i+1}] \not\subset X$ , let  $z_i = \min\{x \in X: x > x_i \text{ and } \sigma(x_i) \cdot \sigma(x) \leq 0\}$ . Set  $z_0 = \min\{x \in X\}$ ,  $z_{n+1} = \max\{x \in X\}$ . (It is possible that  $z_n = z_{n+1}$ , although this will present no difficulty as will become clear later).

Now we construct a new reference set  $Y = \{y_1, \dots, y_{n+1}\} \subset X$  of distinct points so that the function  $\sigma(x)$  still alternates in sign, that is  $\sigma(y_i) \cdot \sigma(y_{i+1}) < 0$  for  $i = 1, \dots, n$ ,  $|\sigma(y_i)| \geq |\sigma(x_i)|$  for  $i = 1, \dots, n + 1$ , and for some  $k$ ,  $k = 1, \dots, n + 1$   $|\sigma(y_k)| = \|f - p\|$ . We first find a trial reference set  $\tilde{Y} = \{\tilde{y}_1, \dots, \tilde{y}_{n+1}\}$  by choosing  $\tilde{y}_i$  to be a point in  $[z_{i-1}, z_i] \cap X$  where  $\sigma(x) \cdot (\text{sign}(\sigma(x_i)))$  attains its

maximum on  $[z_{i-1}, z_i] \cap X$ . If there is a  $k$  such that  $|\sigma(\tilde{y}_k)| = \|f-p\|$ , then we set  $Y = \tilde{Y}$ . Otherwise, let  $y^*$  be a point in  $X$  so that  $|\sigma(y^*)| = \|f-p\|$ . We form  $Y$  by adding  $y^*$  to  $\tilde{Y}$  and removing one of the  $\tilde{y}_i$ 's so that  $Y$  is still ordered and  $\sigma(x)$  alternates in sign on  $Y$ . This may be accomplished as follows. If  $y^* \in (\tilde{y}_k, \tilde{y}_{k+1}) \cap X$ , choose  $j$  to be either  $k$  or  $k+1$  so that  $\sigma(\tilde{y}_j)$  is of the same sign as  $\sigma(y^*)$ . Then replace  $\tilde{y}_j$  by  $y^*$  to form  $Y$ . If  $y^* < \tilde{y}_1$ , eliminate  $\tilde{y}_{n+1}$  (by the way  $\tilde{y}_1$  was chosen, it must be the case that  $\text{sign}(\sigma(y_1)) = -\text{sign}(\sigma(y_{n+1}))$ ). Similarly, if  $y^* > \tilde{y}_{n+1}$ , eliminate  $\tilde{y}_1$ .

If the new reference set  $Y$  is the same as the reference set from the last iteration,  $R$ , then it is necessarily true that  $|\lambda| = \|f-p\|$ , and from the Characterization theorem,  $p$  must be the best approximation to  $f$  from  $S$ , and we terminate the algorithm. If  $Y \neq R$ , then we iterate again, substituting our new reference set  $Y$  for the old reference set  $R$ . The exchange procedure outlined above insures that as long as the  $\lambda$  determined by the initial reference set is non-zero then at each successive iteration, the magnitude of the present  $\lambda$  is strictly greater than the magnitude of the  $\lambda$  in the preceding iteration (see [3]).

It can be shown [6, p. 108] that the above algorithm does converge (uniformly) to the best approximation to  $f$  from  $S$ . In fact, under stronger hypotheses, the convergence is quadratic [6, p. 111].

We may extend these results by forcing the approximation functions to satisfy certain side conditions. We now consider two such extensions. First, we consider imposing interpolatory constraints on the approximating functions and their derivatives, then we shall consider restricting the range of the approximating functions. In both cases, if we make strong enough assumptions on the structure of the subspace  $S$ ,

the questions of existence, uniqueness and continuous dependence on the data can be affirmatively answered (see [5] for interpolatory constraints; see [8] for restricted range). Moreover, in both cases, the Remes algorithm can be extended to effectively compute best approximations from these new approximation classes.

In order to consider the interpolatory constraint problem we must first define an extended Haar system. We follow the notation in [5]. Let  $\{u_i(t)\}_{i=1}^n$  be a family of functions in  $C[a, b]$  (with each sufficiently differentiable so that what follows is well defined) and  $\{t_i\}_{i=1}^n \subset [a, b]$  be such that  $a \leq t_1 \leq t_2 \leq \dots \leq t_n \leq b$ . Define

$$u^*(\begin{smallmatrix} 1, \dots, n \\ t_1, \dots, t_n \end{smallmatrix}) = \begin{vmatrix} \tilde{u}_1(t_1) & \tilde{u}_1(t_2) & \dots & \tilde{u}_1(t_n) \\ \vdots & & & \vdots \\ \tilde{u}_n(t_1) & \tilde{u}_n(t_2) & \dots & \tilde{u}_n(t_n) \end{vmatrix},$$

where for fixed  $j$

$$\tilde{u}_i(t_j) = \begin{cases} u_i(t_j) & \text{if } t_{j-1} < t_j \\ u_i^{(\mu)}(t_j) & \text{if } t_{j-\mu} = t_{j-\mu+1} = \dots = t_j \end{cases}$$

$1 \leq i \leq n$ . The functions  $\{u_i(t)\}_{i=1}^n$  will be called an extended Chebyshev system of order  $v$  on  $[a, b]$  provided  $u_i \in C^{v-1}[a, b]$ ,  $i = 1, \dots, n$ , and

$$u^*(\begin{smallmatrix} 1, \dots, n \\ t_1, \dots, t_n \end{smallmatrix}) > 0$$

for all choices  $t_1 \leq t_2 \leq \dots \leq t_n$ ,  $t_i \in [a, b]$ , where equality occurs in groups of at most  $v$  consecutive  $t_i$  values. An  $n$ -dimensional subspace

$M$  of  $C[a, b]$  is said to be an extended Haar subspace of order  $v$ ,  $v \leq n$ , provided that there exists a basis for  $M$  which is an extended Chebyshev system of order  $v$  on  $[a, b]$ .

We may state the interpolatory constraint problem as follows. Let  $M$  be an  $n$ -dimensional extended Haar subspace of order  $v$  (on  $[a, b]$ ). Let  $\{x_i\}_{i=1}^k \subset X \subset [a, b]$  satisfy  $x_1 < x_2 < \dots < x_k$ , and let  $\{m_i\}_{i=1}^k$  be a set of positive integers such that  $\max_{1 \leq i \leq k} (m_i) - 1 \leq v$ , and  $\sum_{i=1}^k m_i = m < n$ . We assume  $X$  contains at least  $n - m + k + 1$  points. Let  $\{a_{ij}\}_{i=1, j=0}^{k, m_i-1}$  be a set of  $m$  real numbers and define

$$S = \{p \in M : p^{(j)}(x_i) = a_{ij}, 1 \leq i \leq k \text{ and } 0 \leq j \leq m_i - 1\}.$$

The interpolating constraint problem then is to find best approximations to functions in the class

$$\tilde{C}(x) = \{f \in C(x) : f(x_i) = a_{i0}, 1 \leq i \leq k\}$$

from the class  $S$ .

The Remes algorithm may be extended in the following manner to compute interpolatory approximations. Let  $f \in \tilde{C}(x)$ ,  $\{x_i\}_{i=1}^k$ ,

$\{m_i\}_{i=1}^k$  and  $\{a_{ij}\}_{i=1, j=0}^{k, m_i-1}$  be as above. Let  $R = \{t_1, \dots, t_{n+1}\} \subset X$

be such that  $t_1 \leq t_2 \leq \dots \leq t_{n+1}$ , and each  $x_i$  is repeated in  $R$   $m_i$

times. Moreover, if  $t_j \in R \sim \bigcup_{i=1}^k x_i$ , then  $t_{j-1} < t_j < t_{j+1}$ . Let

$\{j_\ell\}_{\ell=1}^{n-m+1}$  be such that  $\{t_{j_\ell}\}_{\ell=1}^{n-m+1} = R \sim \bigcup_{i=1}^k x_i$ . It can be shown (see

[5]) that there exist a unique  $p \in S$  and a real number  $\lambda$  satisfying

$$f(t_{j_\ell}) - p(t_{j_\ell}) = (-1)^{j_\ell} \lambda, \ell = 1, 2, \dots, n-m+1,$$

and

$$p^{(j)}(x_i) = a_{ij}, j = 0, 1, \dots, m_j - 1, i=1, \dots, k.$$

We assume that  $R$  is initially chosen so that  $|\lambda| > 0$ .

Next we perform an exchange on the reference set  $R$  to form a new reference set  $Y \subset X$  with the properties that each  $x_i$  is repeated in  $Y$   $m_i$  times, that there is a  $y^* \in Y$  such that  $|f(y^*) - p(y^*)| = \|f-p\|$ , and if  $\{\tilde{t}_{i_\ell}\}_{\ell=1}^k = Y \sim \bigcup_{j=1}^k x_j$  then

$$\text{sign}\{f(\tilde{t}_{i_\ell}) - p(\tilde{t}_{i_\ell})\} = (-1)^q \text{sign}\{f(\tilde{t}_{i_1}) - p(\tilde{t}_{i_1})\}, \ell = 1, \dots, n-m+1,$$

where  $q = i_\ell - i_1$ . After the exchange, if  $R \neq Y$  we iterate again using the reference set  $Y$  in place of  $R$ . If  $R = Y$  then the present  $p$  is the best approximation to  $f$  from  $S$ , and the algorithm is terminated. For a description of the exchange and a proof that this algorithm converges uniformly, see [5].

We now consider uniform restricted range approximations. For a given  $f \in C(X)$  we assume that we have two extended real valued functions  $u(x)$  and  $\ell(x)$  defined on  $X$  which satisfy the following requirements:

- (i)  $\ell(x)$  may assume the value  $-\infty$  but never  $+\infty$ .
- (ii)  $u(x)$  may assume the value  $+\infty$  but never  $-\infty$ .
- (iii)  $A = \ell^{-1}(-\infty)$  and  $B = u^{-1}(+\infty)$  are open sets in  $X$ .
- (iv)  $\ell(x)$  is continuous on  $X \sim A$ .
- (v)  $u(x)$  is continuous on  $X \sim B$ .
- (vi)  $\ell(x) \leq f(x) \leq u(x)$  for all  $x \in X$ .
- (vii)  $\ell(x) < u(x)$  for all  $x \in X$ .

Let  $S$  be an  $n$ -dimensional Haar subspace of  $C(X)$ . Let  $K = \{p \in S : \ell(x) \leq p(x) \leq u(x) \text{ for every } x \in X\}$ . We say that  $p^*$  is a best restricted range approximation to  $f$  if  $p^* \in K$  and  $\|f - p^*\| = \inf_{p \in K} \|f - p\|$ . Taylor [8] has proved that under the above hypotheses on  $f$ ,  $\ell$ ,  $u$  and  $S$ , and if  $K \neq \emptyset$ , then there exists a unique best restricted range approximation to  $f$ . As long as there is more than a single element in  $K$ , then we have a characterization of the best restricted range approximation to  $f$  as given in the following theorem. Before starting Taylor's characterization theorem, we define for each  $p \in K$  the following subsets of  $X$ .

$$X_{+1} = \{x \in X : f(x) - p(x) = \|f - p\|\}$$

$$X_{-1} = \{x \in X : f(x) - p(x) = -\|f - p\|\}$$

$$X_{+2} = \{x \in X : p(x) = \ell(x)\}$$

$$X_{-2} = \{x \in X : p(x) = u(x)\}$$

$$X_p = X_{+1} \cup X_{-1} \cup X_{+2} \cup X_{-2}.$$

Restricted Range Characterization Theorem [8].  $p^* \in K \subset S$  is a best restricted range approximation to  $f$  if and only if there exist  $n + 1$  distinct points,  $x_0, \dots, x_n$  with  $x_0 < x_1 < \dots < x_n$  in  $X_{p^*}$  satisfying  $\sigma(x_i) = (-1)^{i+1} \sigma(x_1)$ , where  $\sigma(x) = -1$  if  $x \in X_{-1} \cup X_{-2}$  and  $\sigma(x) = +1$  if  $x \in X_{+1} \cup X_{+2}$ .

This characterization is the basis for the following multiple exchange algorithm due to Gimlin, Cavin and Budge [4] which is the analog of the multiple exchange Remes algorithm previously described. We assume  $f \in C(X)$  is fixed and that  $u$ ,  $\ell$ , and  $K$  have been chosen as described above. We also assume that we have an arbitrary reference set  $R = \{x_1, \dots, x_{n+1}\} \subset X$  with  $x_1 < x_2 < \dots < x_{n+1}$  on which no function in  $K$  interpolates  $f$ . Set  $\sigma_i = (-1)^{i+1}$ ,  $f_i = f(x_i)$ ,  $i = 1, \dots, n + 1$ .

We begin the first iteration by finding  $p \in S$  and a real number such that  $p(x_i) + \sigma_i \lambda = \tilde{f}_i$ . We next execute an exchange algorithm which returns a new reference set  $Y \subset X$ , and (possibly) new values for  $\sigma_i$  and  $\tilde{f}_i$ . If our current reference set  $R \neq Y$ , we iterate again substituting  $Y$  for  $R$ . If  $R = Y$ , then our present  $p$  is in  $K$  and satisfies the characterization theorem so that  $p$  is the best restricted range approximation to  $f$ .

We now describe the exchange algorithm. We define the function  $\text{sign}^*$  as follows.

$$\text{sign}^*(f(x) - p(x)) = \begin{cases} \text{sign}(f(x) - p(x)) & \text{if } \ell(x) < f(x) < u(x) \text{ or } f(x) \neq p(x) \\ +1 & \text{if } f(x) = \ell(x) = p(x) \\ -1 & \text{if } f(x) = u(x) = p(x) \end{cases}$$

It can be shown (see [4]) that at each iteration  $\text{sign}^*(f(x_i) - p(x_i)) = -\text{sign}^*(f(x_{i+1}) - p(x_{i+1}))$ ,  $i = 1, \dots, n$ . Set  $z_i = \inf\{x \in [x_i, x_{i+1}] \cap X : \text{sign}^*(f(x) - p(x)) \cdot \text{sign}^*(f(x_i) - p(x_i)) \leq 0\}$ ,  $i = 1, 2, \dots, n$ . Set  $z_0 = \min\{x \in X\}$ ,  $z_{n+1} = \max\{x \in X\}$ . Define  $E_i$ ,  $M_i$ ,  $m_i$  as follows.

$$\hat{E}_i = \max_{x \in [z_{i-1}, z_i] \cap X} \{S_i \cdot (f(x) - p(x)) - |\lambda|\}$$

$$\hat{M}_i = \max_{x \in [z_{i-1}, z_i] \cap X} \{S_i \cdot (u(x) - p(x))\}$$

$$\hat{m}_i = \max_{x \in [z_{i-1}, z_i] \cap X} \{S_i \cdot (\ell(x) - p(x))\}$$

where  $S_i = \text{sign}^*(f(x_i) - p(x_i))$ . Let  $\hat{\gamma}_i = \max\{E_i, M_i, m_i\}$ , in case of equality, choose  $\hat{\gamma}_i$  to be the first largest member of the triple. We define a new trial reference set  $\tilde{Y} = \{\tilde{y}_1, \dots, \tilde{y}_{n+1}\}$  by taking  $\tilde{y}_i$  to be a point  $[z_{i-1}, z_i] \cap X$  at which  $\hat{\gamma}_i$  occurs.

Define  $\gamma = \max\{E, M, m\}$  where

$$E = \max_{x \in X} \{|f(x) - p(x)| - |\lambda|\}$$

$$M = \max_{x \in X} \{p(x) - u(x)\},$$

$$m = \max_{x \in X} \{\ell(x) - p(x)\}.$$

In case of equality, choose  $\gamma$  to be the first largest member of this triple. If for some  $k$ ,  $1 \leq k \leq n$ ,  $\hat{\gamma}_k = \gamma$ , then set  $Y = \hat{Y}$ . Otherwise, let  $y^*$  be a point at which  $\gamma$  occurs, insert  $y^*$  into  $\hat{Y}$  and delete a  $\hat{y}_k \in \hat{Y}$  exactly as was done in the Remes algorithm, preserving sign\* alternation. We now determine our new sets  $\{\sigma_i\}_{i=1}^{n+1}$  and  $\{f_i\}_{i=1}^{n+1}$ . If  $y_i \in Y$  is a point at which  $E$  or  $\hat{E}_i$  occurs, set  $\sigma_i = \text{sign}(f(y_i) - p(y_i))$ , and set  $\hat{f}_i = f(y_i)$ . If  $y_i$  is a point at which  $M$  or  $\hat{M}_i$  occurs, set  $\sigma_i = 0$ , and set  $\hat{f}_i = u(y_i)$ . If  $y_i$  is a point at which  $m$  or  $\hat{m}_i$  occurs, set  $\sigma_i = 0$ , and set  $\hat{f}_i = \ell(y_i)$ . This completes the exchange algorithm.

See Gimlin [4] for a proof that this algorithm does converge to the best restricted range approximation to  $f$ .

As can be done with best uniform approximations, best restricted range approximations can be computed subject to interpolatory constraints. The precise formulation of the problem is completely analogous to that of best uniform approximations with interpolatory constraints. A Remes-like algorithm for computing restricted range approximations subject to interpolatory constraints is available and is analogous to the Remes-like algorithm given above for computing best uniform approximations with interpolatory constraints. For a complete discussion of this problem see [2].

## CHAPTER II

### Two Adaptive Piecewise Polynomial Curve Fitting Algorithms

In this section we present our algorithms for adaptively finding smooth piecewise polynomial uniform approximations. First, we define what we mean by a piecewise polynomial uniform approximation. Suppose the function to be approximated,  $f$ , is defined on  $X$  a compact set of real numbers, and  $[a, b]$  is the smallest interval containing  $X$ . A function  $p$  is a piecewise polynomial approximation to  $f$  of degree  $n$  if there are  $m$  closed intervals,  $[a, x_1], [x_1, x_2], \dots, [x_{m-1}, b] = I_1, I_2, \dots, I_m$ , such that the restriction of  $p$  to  $I_j$  is in  $\Pi_n = \{p : p \text{ is a polynomial of degree less than or equal to } n\}$  for each  $j$ . If  $p$  restricted to  $I_j$  is the best uniform approximation to  $f$  on  $X \cap I_j$  from  $\Pi_n$  for each  $j$ , then  $p$  is said to be the best piecewise polynomial uniform approximation to  $f$  of degree  $n$  (with respect to the partition  $I_1, I_2, \dots, I_m$  of  $[a, b] \cap X$ ). See [1] and [7] for a theoretical study of this problem where  $I_1, \dots, I_m$  are chosen so that  $E_j = \max_{x \in I_j \cap X} \{|f(x) - p(x)|\}$  is the same for all  $j$  (thereby giving the partition containing the minimal number of intervals needed to achieve this error).

However, in many applications of approximation theory, it is also desired to find a smooth approximation. If one adds this constraint to the above problem, then a best smooth piecewise polynomial uniform approximation may not exist having the same error on each subinterval. Even if such an approximation does exist, the number of subintervals in the partition can be unreasonably large. Thus, rather than finding a

best approximation in the above sense, it is reasonable to first stipulate an error bound for  $\|f - p\|$  and a smoothness constraint on the piecewise polynomial. Given these constraints, one then should seek a partition  $I_1, \dots, I_m$  of  $[a, b]$  such that  $m$  is in some sense minimal or small and the piecewise polynomial uniform approximation,  $p$ , to  $f$  with respect to this partition satisfies both the error tolerance and smoothness demands.

This latter approach is what we will follow. Typically, our smoothness constraint will be that  $p \in C^v[a, b]$  where  $v = 0, 1$  or  $2$ . Our adaptive curve fitting algorithms will find smooth piecewise polynomial uniform approximations satisfying a prescribed error tolerance and smoothness requirements by adaptively choosing subintervals. This partition will be chosen in such a way that the error of the best uniform (or restricted range approximation) to  $f$  on the intersection of each subinterval with  $X$  from  $\Pi_n$  subject to interpolatory constraints guaranteeing the desired smoothness is less than the prescribed error tolerance. At the same time, effort is made to keep the number of subintervals as small as possible. In what follows, the points  $\{x_i\}_{i=0}^m$  will be called knots, where  $x_0 = a$ ,  $x_m = b$ , and  $I_j = [x_{j-1}, x_j]$ .

#### Algorithm 1

We assume that  $f \in C(X)$ . We use the following notation:  $n$  denotes the degree of the approximating polynomials. SMTH denotes the number of continuous derivatives desired at the knots ( $\text{SMTH} \leq n$ ); TOL denotes the approximation tolerance we wish our piecewise polynomials to satisfy; and LENGTH denotes an approximate minimum length we will allow for any

subinterval. We assume these values are all given and fixed. We begin by choosing  $\tilde{x}_1 \in X$  to be the largest point in  $X$  which satisfies

$$(1) \quad \tilde{x}_1 - a \geq \text{LNGTH}$$

(2) If  $p_1$  is the best uniform approximation to  $f$  from  $\Pi_n$  on

$$[a, \tilde{x}_1] \cap X, \text{ then } \sup_{x \in [a, \tilde{x}_1] \cap X} |f(x) - p_1(x)| \leq \text{TOL}.$$

If  $\tilde{x}_1 = b$ , then since  $p_1$  is a piecewise polynomial meeting our requirements, the algorithm is successfully terminated. If no such  $\tilde{x}_1$  exists, then the algorithm fails and is terminated with an appropriate error message. Otherwise, we choose  $x_1$  to be the largest extreme point of  $f(x) - p_1(x)$  in  $(a, \tilde{x}_1) \cap X$  (which is easily available to us from the Remes algorithm).

We continue by finding successive intervals  $[x_1, x_2], [x_2, x_3], \dots, [x_{m-1}, b]$ , and polynomial approximations  $p_2, p_3, \dots, p_m \in \Pi_n$  to  $f$  so that  $p_i^{(j)}(x_i) = p_{i+1}^{(j)}(x_i)$  for  $i = 1, 2, \dots, m-1$ ,  $j = 0, 1, \dots, \text{SMTH}$ , and so that  $\sup_{x \in [x_{i-1}, x_i] \cap X} |f(x) - p_i(x)| \leq \text{TOL}$ . This is accomplished as follows.

Suppose we have determined the subintervals  $[a, x_1], [x_1, x_2], \dots, [x_{i-2}, x_{i-1}]$ , and the approximations  $p_1, p_2, \dots, p_{i-1}$ . Assume further that  $b - x_{i-1} \geq \text{LNGTH}$ . We now determine an  $x_i$  and a  $p_i$  meeting the above requirements. We begin by choosing  $\tilde{x}_i \in X$  to be the largest point in  $X$  which satisfies

$$(1) \quad \tilde{x}_i - x_{i-1} \geq \text{LNGTH},$$

(2) If  $p_i$  is the best approximation to  $f$  from  $\Pi_n$  on  $[x_{i-1}, \tilde{x}_i] \cap X$  subject to the constraint that  $p_i^{(j)}(x_{i-1}) = p_{i-1}^{(j)}(x_{i-1})$ ,

$$j = 0, 1, \dots, \text{SMTH}, \text{ then } \sup_{x \in [x_{i-1}, \tilde{x}_i] \cap X} |f(x) - p_i(x)| \leq \text{TOL}.$$

If  $\tilde{x}_i = b$ , we set  $x_i = \tilde{x}_i = b$ , and the algorithm is successfully terminated. If no such  $\tilde{x}_i$  exists, the algorithm fails and is terminated with an appropriate error message. Otherwise, we choose  $x_i$  to be the largest extreme point of  $f(x) - p_i(x)$  in  $(x_{i-1}, \tilde{x}_i) \cap X$ .

Finally, we consider the special case where  $b - x_{i-1} < \text{LNGTH}$ . In this case we change  $x_{i-1}$  to some  $\hat{x}_{i-1} \in X$ , where  $x_{i-2} < \hat{x}_{i-1} < x_{i-1} < b$ . So that  $b - \hat{x}_{i-1} \geq \text{LNGTH}$ . Specifically, we choose  $\hat{x}_{i-1}$  to be a point in  $X$  closest to  $(b - x_{i-2})/2$  which satisfies

$$(1) \quad b - \hat{x}_{i-1} \geq \text{LNGTH},$$

$$(2) \quad \text{If } p_i \text{ is the best approximation to } f \text{ from } \Pi_n \text{ on } [\hat{x}_{i-1}, b] \cap X \\ \text{subject to the constraint that } p_i^{(j)}(\hat{x}_{i-1}) = p_{i-1}^{(j)}(\hat{x}_{i-1}) \text{ for} \\ j = 0, 1, \dots, \text{SMTH}, \text{ then } \sup_{x \in [\hat{x}_{i-1}, b] \cap X} |f(x) - p_i(x)| \leq \text{TOL}.$$

Again, if we can find such an  $\hat{x}_{i-1}$ , then the algorithm is successfully terminated. If not, the algorithm fails and is terminated with an appropriate error message.

### Algorithm 2

We assume that  $f$ ,  $\ell$ , and  $u \in C(X)$  and satisfy the hypotheses necessary for restricted range approximation. Let  $n$ ,  $\text{SMTH}$  and  $\text{LNGTH}$  be as in Algorithm 1. If  $\text{TOL}$  is the tolerance we wish the approximation to satisfy, then we make the additional assumption that  $\ell$  and  $u$  have been chosen so that if  $p$  is any function satisfying  $\ell(x) \leq p(x) \leq u(x)$  for all  $x \in X$ , then  $\|f - p\| \leq \text{TOL}$ . We begin by choosing  $\tilde{x}_1 \in X$  to be the

largest point in  $X$  satisfying

- (1)  $\tilde{x}_1 - a \geq \text{LNGTH}$ ,
- (2) There is a best restricted range approximation,  $p_1(x)$ , to  $f$  from  $\Pi_n$  on  $[a, \tilde{x}_1] \cap X$ .

If  $\tilde{x}_1 = b$ , then the algorithm is successfully terminated. If no such  $x_1$  exists, the algorithm fails and is terminated with an appropriate error message. Otherwise, we choose  $x_1$  to be the largest extreme point of  $f(x) - p_1(x)$  in  $(a, \tilde{x}_1) \cap X$ .

Proceeding analogously to Algorithm 1, in general, if  $b - x_{i-1} \geq \text{LNGTH}$ , we choose  $\tilde{x}_i \in X$  to be the largest point in  $X$  so that

- (1)  $\tilde{x}_i - x_{i-1} \geq \text{LNGTH}$ ,
- (2) There is a best restricted range approximation,  $p_i(x)$ , to  $f$  from  $\Pi_n$  on  $[x_{i-1}, \tilde{x}_i] \cap X$  satisfying the constraints  $p_i^{(j)}(x_{i-1}) = p_{i-1}^{(j)}(x_{i-1})$ ,  $j = 0, 1, \dots, \text{SMTH}$ .

If  $\tilde{x}_i = b$  the algorithm is successfully terminated. If no such  $\tilde{x}_i$  exists, the algorithm fails and is terminated with an appropriate error message. Otherwise,  $x_i$  is chosen to be the largest extreme point in  $(x_{i-1}, \tilde{x}_i) \cap X$ .

If  $b - x_{i-1} < \text{LNGTH}$ , we choose  $\hat{x}_{i-1} \in X$  to be a point closest to  $(b - x_{i-2})/2$  which satisfies

- (1)  $b - \hat{x}_{i-1} \geq \text{LNGTH}$
- (2) There is a best restricted range approximation,  $p_i(x)$ , to  $f$  from  $\Pi_n$  on  $[\hat{x}_{i-1}, b] \cap X$  satisfying the constraints  $p_i^{(j)}(\hat{x}_{i-1}) = p_{i-1}^{(j)}(\hat{x}_{i-1})$ ,  $j = 0, 1, \dots, \text{SMTH}$ .

If we can find an  $\hat{x}_{i-1}$ , the algorithm is successfully terminated; if not, the algorithm fails and an appropriate error message is printed.

#### Remarks

We first note that in algorithm 1 it is in general not necessary to compute the best approximation on a particular subinterval in order to conclude that the subinterval is too long. Indeed, at each iteration of the Remes algorithm we obtain a lower bound for the error of the best approximation on this particular subinterval, namely  $|\lambda|$ . Consequently, if during some iteration of the Remes algorithm  $|\lambda| > \text{TOL}$ , we may conclude that the present subinterval is too large and terminate the Remes algorithm.

In our implementation of these algorithms, the  $\hat{x}_i$  are chosen as follows. At each step of this iterative procedure we will let  $\tilde{a}$  be the current largest point in  $X$  such that requirements (1) and (2) of the appropriate algorithm are satisfied on  $[x_{i-1}, \tilde{a}] \cap X$ , and we will let  $\tilde{b}$  be the current smallest point in  $X$  such that  $\tilde{b} > \tilde{a}$  and requirement (2) of the appropriate algorithm fails to be satisfied. We initialize this process by computing (or attempting to compute) the appropriate best approximation on  $[x_{i-1}, b] \cap X$ . If this approximation satisfies requirement (2), then we set  $\tilde{x}_i = b$  and we are done. If the approximation fails to satisfy (2), we set  $\tilde{b} = b$ . Next, let  $t = \min\{x \in X : x - x_{i-1} \geq \text{LNGTH}\}$ . If the approximation on  $[x_{i-1}, t] \cap X$  fails to satisfy (2) then the algorithm cannot meet the desired accuracy and fails. Otherwise, we set  $\tilde{a} = t$ .

In general, we proceed as follows. We let  $t = \inf\{x \in X : (\tilde{b} - \tilde{a})/2 \leq x < \tilde{b}\}$ . If this set is empty, we set  $t = \sup\{x \in X : a \leq x \leq (\tilde{b} - \tilde{a})/2\}$ .

If  $t = \tilde{a}$  then this procedure has converged and we set  $\tilde{x}_i = t = \tilde{a}$ . Otherwise, we compute (or attempt to compute) the appropriate approximation on  $[x_{i-1}, t] \cap X$ . If this approximation satisfies (2) then we set  $\tilde{a} = t$ . If this approximation fails to satisfy (2) then we set  $\tilde{b} = t$ . We continue this process until  $\tilde{b} - \tilde{a}$  is less than some prescribed tolerance, at which point we accept  $\tilde{a}$  as a good approximation to  $\tilde{x}_i$  and terminate this procedure. We compute the  $\hat{x}_i$  in a manner analogous to the above.

In an attempt to accelerate the convergence of the above scheme in curve fitting algorithm 1, we have tried to take advantage of the fact that corresponding to each  $\tilde{a}$  we know the error of approximation on  $[x_{i-1}, \tilde{a}] \cap X$ , call it SMLERR, and corresponding to each  $\tilde{b}$  we know a lower bound for the error of approximation on  $[x_{i-1}, \tilde{b}] \cap X$ , call it BIGERR. We change the above scheme by first setting  $\alpha = (\text{BIGERR} - \text{TOL}) / (\text{BIGERR} - \text{SMLERR})$  and then setting  $t = \inf\{x \in X : \alpha\tilde{a} + (1-\alpha)\tilde{b} \leq x \leq \tilde{b}\}$  or, if this set is empty, we set  $t = \sup\{x \in X : \alpha\tilde{a} + (1-\alpha)\tilde{b} \leq x \leq \tilde{b}\}$  in the general iteration described above. Hence, if SMLERR is very close to TOL,  $t$  will be chosen close to  $\tilde{a}$ . Our numerical experience has shown that this procedure only works well when approximating uniformly smooth functions and that this algorithm cannot be significantly improved by allowing the Remes algorithm to run to completion in order to obtain the true error of approximation for BIGERR.

If  $f(x)$  is differentiable on some interval and  $x_0$  is an interior relative extreme point of  $f(x) - p(x)$  in this interval, then  $f'(x_0) = p'(x_0)$ . Hence, by "backing off" from  $\tilde{x}_i$  to  $x_i$  we hope to force the first derivative of the approximating polynomial and the first

derivative of the function being approximated to agree closely at the knots, thereby dampening oscillatory effects otherwise prevalent in this type of approximation when we require the approximation to be at least continuously differentiable. Our numerical experience has shown that this procedure is crucial in order to obtain reasonable approximations. If we only want our approximations to be continuous at the knots, then this procedure accomplishes nothing and should be overridden.

In our implementation of these algorithms, we have always assumed that  $X$  is a finite set of points. Because the Remes algorithm can only find approximations on sets of at least  $n + 1$ -SMTH points, one of the uses of the parameter  $LNGTH$  is to insure that every set on which we approximate contains at least  $n + 1$ -SMTH points.

If we set  $l(x) = f(x) - TOL$  and  $u(x) = f(x) + TOL$  in algorithm 2, the approximations obtained from algorithm 2 are the same as those obtained from algorithm 1. However, for a broad class of approximation problems, the additional computational complexity of algorithm 2 justifies the use of the simpler algorithm 1. We will discuss this aspect in greater detail in the next section. Algorithm 2 is particularly suited to problems where the desired tolerance varies over the length of the interval. In the next section we will describe a strategy which essentially allows us to weight data points which contain significant levels of noise by an appropriate choice of  $l(x)$  and  $u(x)$ .

### CHAPTER III

#### Numerical Results

Algorithms 1 and 2 have been implemented as FORTRAN programs and have been tested on Colorado State University's CDC CYBER 172. Care has been taken to make the programs reasonably efficient and consistent with each other so that CPU time comparisons should be meaningful. As examples, the functions  $e^{|x|}$  on  $[-1, 1]$ ,  $|\sin(x)|$  on  $[-\pi, \pi]$  and  $\sqrt{x}$  on  $[0, 2]$  were approximated on 200 equally spaced points with  $TOL = .01$ ,  $SMTH = 2$ , and  $N = 6$ . When using algorithm 2 we chose  $u(x) = f(x) + TOL$  and  $l(x) = f(x) - TOL$ . Each of these examples is most difficult to approximate by polynomials near  $x = 0$ . Consequently, the algorithms' ability to automatically decrease the length of the subintervals near  $x = 0$  and then to recover by lengthening them for  $x > 0$  is tested. As noted above, the approximations computed by the two algorithms should (and do) agree up to machine accuracy. Below is a table listing knot locations (subinterval endpoints) and the CPU time in seconds used by each algorithm to compute the piecewise polynomial approximation.

		$e^{ x }$	$ \sin(x) $	$\sqrt{x}$
CP TIMES	Algorithm 1	.979	1.095	.876
	Algorithm 2	1.518	1.539	1.367
Knot Locations		-1.0	-3.14	0.0
		-.347	-.268	.0804
		-.0653	.0474	.231
		.0452	.205	.352
		.196	.458	.814
		.437	.868	2.0
		1.0	1.53	-----
		-----	3.14	-----

We have found that in order for these algorithms to be stable, it is crucial that when computing smooth approximations ( $SMTH > 0$ ), the derivative of the approximating polynomial agree closely with the slope (or approximate derivative) of the function being approximated at the right endpoint of every subinterval except the last. When approximating on a dense enough subset of an interval, the strategy of choosing  $x_i$  to be the last interior extreme point of  $f - p$  given in the description of the algorithms seems to be sufficient. However, for more widely spaced points, the true last interior extreme point of  $f - p$ ,  $x^*$ , generally is not in  $X$  and often, the closest point in  $X$  to  $x^*$ , which is found by the Remes algorithm, is not a point at which  $p'$  agrees very closely with the slope of  $f$ . Our numerical experience has shown that by setting  $\tilde{f}'(x)$  to be the derivative of the centered quadratic interpolation of  $f$  at  $x$  and choosing  $x_i$  to be the largest extreme point returned by Remes such that  $\{\tilde{f}'(x) - p'(x)\}$  is a minimum we obtain a considerably more stable algorithm.

For some applications it might be preferable to be able to choose ahead of time a particular knot location and to specify interpolatory constraints at these knots. For example, when approximating  $|\sin(x)|$  it might be advantageous to force a knot to be located at  $x = 0$  and to

require  $p(0) = 0$ ,  $p'(0) = -1$  as  $x$  approaches 0 from the left, and  $p'(0) = -1$  as  $x$  approaches 0 from the right. The modifications to the algorithms as given necessary to accomplish this would not be extensive or difficult.

Because uniform approximations weight each data point equally, they are particularly suited for approximating precise mathematical functions or for approximating data with noise levels that are very small relative to the desired accuracy. However, when these algorithms are used to approximate functions containing considerable levels of noise, the approximations tend to follow the noise patterns more than is desirable so that near abrupt changes in the data, the approximations tend to begin to oscillate and generally it requires several subintervals to dampen these oscillations.

For example, we were given some experimental data involving the release of bitumen from oil shale heated to a constant temperature as a function of time. Because relatively few data points were available, we filled in the gaps between the data points by linear interpolation so that we approximated on 200 equally spaced points. Figure 1 is a plot of the data being approximated, the constraining curves  $u(x)$  and  $l(x)$ , and the approximation generated by algorithm 2 where  $N = 6$ ,  $SMTH = 2$ ,  $TOL = 2.5$ ,  $u(x) = f(x) + TOL$ , and  $l(x) = f(x) - TOL$ . Notice that as oscillations appear in the data between times of 22 and 34 minutes, greatly exaggerated oscillations appear in the approximation.

RESTRICTED RANGE 75 GAL/TON TEMP=425 BITUMEN  
 N=6, NSMTH=2, TOL=2.500. Knots are indicated by  $\square$ .

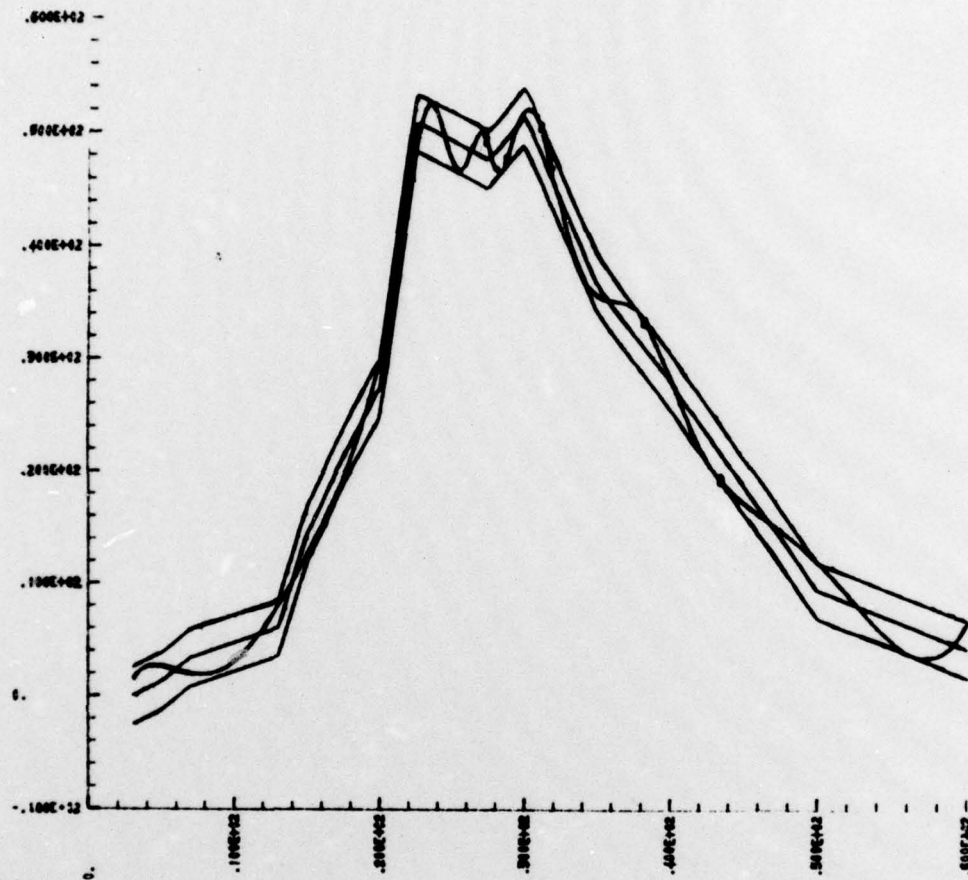


Figure 1

In order to dampen this effect, we have found that by appropriately choosing  $u(x)$  and  $l(x)$  we can improve upon this approximation. Essentially, we choose  $u(x)$  and  $l(x)$  such that for each  $x \in X$ ,  $\max\{u(x) - f(x), f(x) - l(x)\} = \text{TOL}$ , and  $u(x)$  and  $l(x)$  are as smooth as is reasonably possible. Thus, the data points are scattered inside a band of varying width, but neither boundary is further than TOL from a data point. When one data point differs sharply from those on either side of it, we appropriately choose either  $u(x)$  or  $l(x)$  to be close to

this data point. The effect of this choice of  $u$  and  $l$  is to tend to force the approximation toward the more "believable" side of the data which results in fewer oscillatory problems. Figure 2 is a plot of the same example shown in Figure 1, but with  $u(x)$  and  $l(x)$  chosen as above.

RESTRICTED RANGE 75 GAL/TON TEMP=425 BITUMEN  
N=6, NSMTH=2, TOL=2.500. Knots are indicated by  $\square$ .

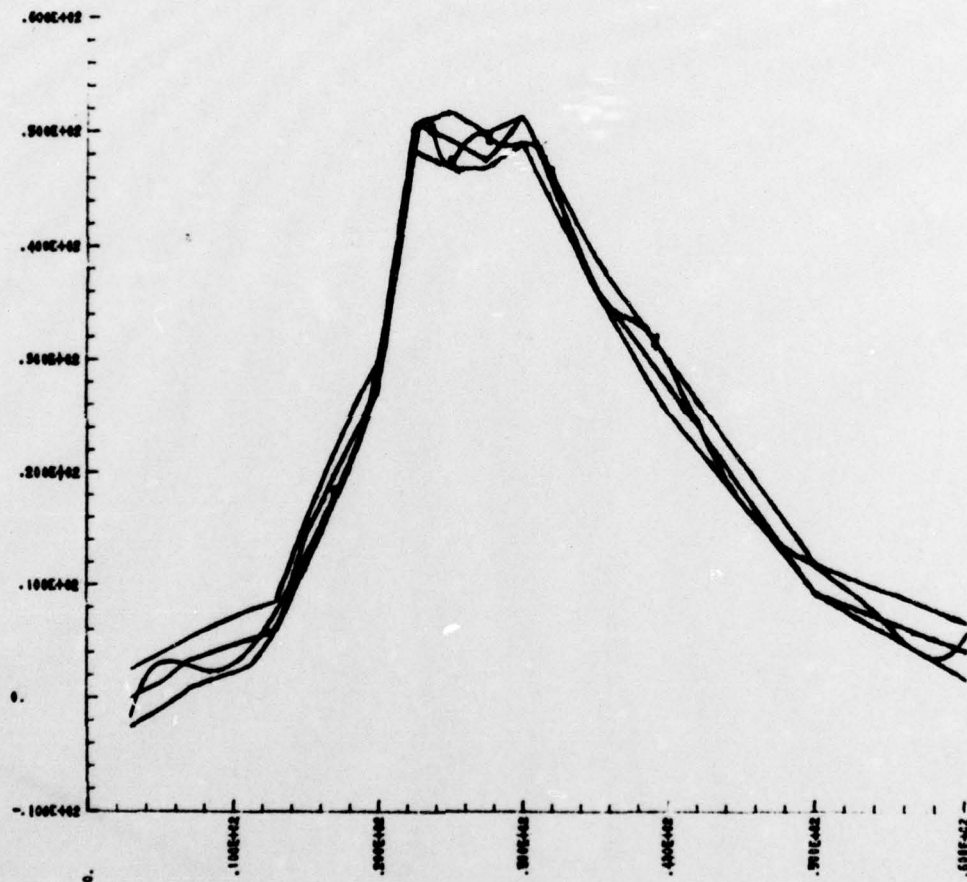


Figure 2

If a technician collecting data knows the general trend his data should follow, he can set  $u(x)$  and  $l(x)$  accordingly, forcing the approximation to lie within theoretical bounds instead of just

prescribing an error tolerance. Or, if he knows that part of his data is highly accurate relative to the rest, he can easily vary the tolerance he is willing to accept accordingly. If he finds that the approximation generated using one set of restraining curves allows the approximation to have undesirable characteristics, generally he can alter these characteristics by appropriately changing  $u(x)$  and  $l(x)$ .

There is no requirement inherent in these algorithms for the data points to be equally spaced. By adding more points in regions where there are quick variations in the function being approximated, smoother approximations result due to the fact that the algorithm will find more optimal right end points as they "back off" from the  $\tilde{x}_i$ .

Because  $L^2$  approximations minimize the effects of normally distributed random noise, we are experimenting with altering algorithm 1 by computing best  $L^2$  approximations in each subinterval instead of best uniform approximations, the theory for best  $L^2$  restricted range approximations is very difficult and has not been developed; therefore, we cannot construct the analog to algorithm 2 using best  $L^2$  approximations. We will report on this work in a future paper.

Recently, John Rice has written several papers in which he describes an adaptive piecewise polynomial algorithm which uses local Hermite interpolation instead of uniform approximation on each subinterval to obtain each polynomial piece (see for example [9]). His algorithm requires that the first SMTH derivatives or approximate derivatives of  $f$  be available in order to compute approximations which have SMTH continuous derivatives. Rice's adaptive strategy also differs from ours; he uses a bisection strategy which can be described as follows. First,

compute the Hermite interpolating polynomial to  $f$  on  $[a, b]$ . That is, compute the polynomial which interpolates  $f$  and its first SMTH derivatives at both endpoints, as well as interpolating  $f$  at  $k$  evenly distributed points in  $(a, b)$ , where  $k = \text{DEGREE} - 2 \cdot \text{SMTH} - 1$ , and DEGREE is the degree of the approximating polynomial. Next, measure the error of approximation using any preselected  $L^p$  norm ( $p \geq 1$ ). If the error is less than TOL, the algorithm terminates, otherwise bisect  $[a, b]$  into two subintervals and check to see if the Hermite interpolating polynomial on  $[a, (a + b)/2]$  differs from  $f$  in norm by no more than TOL. Meanwhile, place the subinterval  $[(a + b)/2, b]$  in a "stack" to be processed later. Continue bisecting subintervals, placing the right half of the interval on the top of the stack to be processed later and check the left half to see if the error of approximation by the Hermite interpolating polynomial is less than TOL. When a subinterval and its associated approximation are found which meet the desired tolerance, we accept this approximation as a "piece" of the piecewise polynomial approximation and continue the algorithm by removing the subinterval which is currently at the top of the stack and repeating the above on it unless the stack is empty, at which point terminate the algorithm.

Rice's routine requires the value of the function and its derivatives at very many points in  $[a, b]$ , even though it may not actually access these values. His algorithm is particularly suited, then, for approximating precise mathematical functions for which this data is readily available. Our algorithms seem more suitable for data reduction in the sense that instead of storing a large number of data points, the user could instead store the coefficients of several polynomials

resulting in a savings of storage space. Our approximations seem preferable for this purpose since our algorithms do not require any information about derivatives, and no special configuration of the data is needed for our adaptive strategy to work.

By using interpolatory constraints on both endpoints of each subinterval, and interpolating the values of the derivative of  $f$  at these points, we could modify our algorithms to use a bisection adaptive strategy. However, in order to take greatest advantage of our use of the best uniform approximation operator as opposed to the Hermite interpolation operator, we felt that it was preferable to use the majority of our degrees of freedom for inside subintervals instead of using them up on interpolation requirements at the endpoints. Essentially, in our algorithms we traded a somewhat faster run time (since the uniform approximation operator is computationally very slow relative to the Hermite interpolation operator) for fewer knots, the freedom from needing to know derivative information, and in algorithm 2, greater control of the characteristics of the approximations.

# REFERENCES

- [1] H. G. Burchard and D. F. Hale, Piecewise Polynomial Approximation on Optimal Meshes, *J. Approx. Theory*, 14 (1975), 128-147.
- [2] B. L. Chalmers, The Remez Exchange Algorithm for Approximation with Linear Restrictions, *Trans. Amer. Math. Soc.*, 222 (1976), 103-131.
- [3] E. W. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966.
- [4] D. R. Gimlin, R. K. Cavin, III, and M. C. Budge, Jr., A Multiple Exchange Algorithm for Calculations of Best Restricted Approximations, *SIAM J. Numer. Anal.*, Vol. 11, No. 2, April, 1974.
- [5] H. L. Loeb, D. G. Moursund, L. L. Schumaker, G. D. Taylor, Uniform Generalized Weight Function Polynomial Approximation with Interpolation, *SIAM J. Numer. Anal.*, Vol. 6, No. 2, June, 1969.
- [6] Günter Meinardus, *Approximation of Functions: Theory and Numerical Methods*, Springer-Verlag, New York, 1967.
- [7] T. Pavlidis and A. P. Maika, Uniform Piecewise Polynomial Approximation with Variable Joints, *J. Approximation Theory*, 12 (1974), 61-69.
- [8] G. D. Taylor, Approximation by Functions Having Restricted Ranges III, *J. Math. Anal. and Appl.* 27 (1969), 241-248.
- [9] J. R. Rice, An Algorithm for Adaptive Piecewise Polynomial Approximation, to appear.

18 (19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER <b>AFOSR-TR-77-0643</b>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER <b>9</b>	
4. TITLE (and Subtitle)  <b>ADAPTIVE PIECEWISE POLYNOMIAL UNIFORM APPROXIMATION</b>		5. TYPE OF REPORT & PERIOD COVERED <b>Interim rept.</b>	
7. AUTHOR(s) <b>Joseph A. Hull</b>		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS <b>Department of Mathematics Colorado State University Fort Collins, Colorado 80523</b>		8. CONTRACT OR GRANT NUMBER(s) <b>AFOSR-76-2878</b>	
11. CONTROLLING OFFICE NAME AND ADDRESS <b>AFOSR/NM Bldg. 410, Bolling AFB Washington, DC 20332</b>		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS <b>61102F</b>	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE <b>March 1977</b>	
		13. NUMBER OF PAGES <b>27</b>	
		15. SECURITY CLASS. (of this report) <b>UNCLASSIFIED</b>	
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)  <b>Approved for public release, distribution unlimited.</b>			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <b>Piecewise polynomial, adaptive approximation, uniform approximation with interpolatory constraints, restricted range approximation with interpolatory constraints.</b>			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <b>In this thesis we present two algorithms for the computation of smooth piecewise polynomial approximations which are adaptive in nature and which are based upon results from uniform approximation theory. We also present a discussion of existing algorithms for computing uniform approximations, uniform approximations with interpolatory constraints, and restricted range uniform approximation. Finally, we report on our numerical testing of our two algorithms, and offer suggestions for their effective use.</b>			